

Numerical Strategies of Computing the Luminosity Distance

De-Zi Liu¹, Cong Ma¹, Tong-Jie Zhang^{*1,2} and Zhi-Liang Yang¹

¹*Department of Astronomy, Beijing Normal University, Beijing 100875, P. R. China*

²*Center for High Energy Physics, Peking University, Beijing 100871, P. R. China*

ABSTRACT

We propose two efficient numerical methods of evaluating the luminosity distance in the spatially flat Λ CDM universe. The first method is based on the Carlson symmetric form of elliptic integrals, which is highly accurate and can replace numerical quadratures. The second method, using a modified version of Hermite interpolation, is less accurate but involves only basic numerical operations and can be easily implemented. We compare our methods with other numerical approximation schemes and explore their respective features and limitations. Possible extensions of these methods to other cosmological models are also discussed.

Key words: Cosmology: miscellaneous — distance scale — methods: numerical

1 INTRODUCTION

The computation of cosmological distances naturally arises in the study of cosmology, for example the luminosity distance d_L in the analysis of type Ia supernova (SNIa) data and the angular diameter distance d_A in the study of gravitational lensing. These distances depend on the underlying cosmological model and their parameters. Therefore they are useful as cosmological tests. As a result, accurate and efficient numerical algorithms of evaluating these distances become a necessity for the practitioners of cosmological research.

The analytical form of the cosmological distances can be derived from the solution of the Friedmann equation, an ordinary differential equation involving the scale factor a as a function of cosmic time t . Therefore, formulae for the distances usually involve an integral over the expansion history expressed in terms of the redshift z and cosmological parameters. In general, the integrations can be evaluated numerically by quadrature algorithms. However, numerical quadratures tend to be computationally heavy when high accuracy is desired.

In the presence of this performance issue, it is advantageous to develop algorithms that are restricted to specific cosmological models but are otherwise more efficient than general-purpose quadratures. For the spatially flat Λ CDM model, efficient algorithms for the luminosity distance have been proposed by Pen (1999, henceforward Pen99) and Wickramasinghe & Ukwatta (2010, henceforward WU10).

In this paper we propose two different numerical methods for the luminosity distance also in the context of the

spatially flat Λ CDM universe. The methods are presented in Sections 2 and 3 respectively. In Section 4, the performances of these methods are discussed. Finally in Section 5 we discuss some possible extensions to the methods presented in this paper. Throughout the paper we will focus on the luminosity distance only, but the results can be trivially extended to compute the angular diameter distance $d_A = d_L/(1+z)^2$.

2 METHOD I: EVALUATION USING CARLSON SYMMETRIC FORMS

The luminosity distance d_L in the spatially flat Λ CDM universe is given by

$$d_L(z) = \frac{c(1+z)}{H_0} \int_0^z \frac{dt}{\sqrt{\Omega_m(1+t)^3 + \Omega_\Lambda}}, \quad (1)$$

where Ω_m and Ω_Λ are the energy densities corresponding to the matter and cosmological constant respectively: $\Omega_m + \Omega_\Lambda = 1$. Following the notation in Pen99 we introduce the parameter $s = \sqrt[3]{(1 - \Omega_m)/\Omega_m}$ and the change-of-variable $u = 1/t$, and re-write equation (1) as

$$\frac{d_L}{c/H_0} = \frac{1+z}{\sqrt{s\Omega_m}} \left[T(s) - T\left(\frac{s}{1+z}\right) \right], \quad (2)$$

where

$$T(x) = \int_0^x \frac{du}{\sqrt{u^4 + u}}. \quad (3)$$

The integral in equations (1) and (3) are special cases of elliptic integrals. All elliptic integrals can be reduced to several basic forms, the best known of which

* E-mail: tjzhang@bnu.edu.cn

are probably the three kinds of Legendre elliptic integrals (Whittaker & Watson 1969, Chapter 22), with reduction theorems and examples presented in (Abramowitz & Stegun 1972, Chapter 17). In our case it is clearer to express this integral by one of the Carlson symmetric forms $R_F(x_1, x_2, x_3)$, which is defined as

$$R_F(x_1, x_2, x_3) = \frac{1}{2} \int_0^{+\infty} \frac{dt}{\sqrt{(t+x_1)(t+x_2)(t+x_3)}}. \quad (4)$$

Using the reduction theorems¹, it is straightforward to verify that

$$T(x) = 4R_F(m, m+3+2\sqrt{3}, m+3-2\sqrt{3}), \quad (5)$$

where

$$m(x) = \frac{2\sqrt{x^2-x+1}}{x} + \frac{2}{x} - 1.$$

It has been known that the Carlson forms can be computed numerically with high accuracy. Carlson (1979) showed that the computation of R_F can be accomplished iteratively with the error decreasing by a factor of 4^6 after each iteration, therefore achieving fast convergence. Further analysis of the algorithms for R_F and other elliptic integrals can be found in (Carlson 1994), and computer implementation details have been discussed in (Carlson & Notis 1981) and (Press et al. 2007, Chapter 6).

3 METHOD II: APPROXIMATION BY A MODIFIED HERMITE INTERPOLATION

The method presented in Section 2 uses an iterative approach to the computation of R_F . However, there are situations where a closed, approximate formula for the integral in equation (3) is desired. In Pen99 an approximation was obtained using polynomial fit for $T(x)$. In WU10 another method with higher accuracy was proposed. In this section we show how a modified version of Hermite interpolation can lead to a class of approximations similar to that in Pen99.

We intend to approximate equation (3) using only elementary operations, such as polynomial evaluation and n th root where n is a small integer. We note that the behavior of $T(x)$ has several deficiencies. First, the derivative of $T(x)$ becomes singular as $x \rightarrow 0^+$. Second, the domain of $T(x)$ extends to infinity. Either one is detrimental to the approximation using polynomials. However, they can be removed by certain change-of-variables. For example, we can introduce a new function

$$\xi(x) = T^2\left(\frac{1}{x} - 1\right) \quad (6)$$

that has smooth derivatives within the interval $0 < x < 1$ and can be extended to the cases of $x \rightarrow 0^+$ and $x \rightarrow 1^-$. The limiting behaviors of $\xi(x)$ are shown below:

$$\begin{aligned} \xi(0^+) &= A^2, & \xi(1^-) &= 0, \\ \xi'(0^+) &= -2A, & \xi'(1^-) &= -4, \end{aligned} \quad (7)$$

where $A = T(+\infty) = 2.80436 \dots$ is a numerical constant².

Using the end-point conditions in equation (7) one can construct a 3rd-order polynomial, which is a linear combination of the four Hermite basis splines in $[0, 1]$, as a crude approximation with $\sim 20\%$ relative error. This linear combination is unique, allowing no further improvements. However, we note that for realistic values of Ω_m it is not necessary to approximate $\xi(x)$ in the entire interval $[0, 1]$, because the subinterval $[0, \frac{1}{s+1})$ corresponds to the scenario of $z < 0$, i.e. “the future”. Therefore, we can introduce a free parameter x_* as the alternative lower end-point, and only perform the approximation in the subinterval $[x_*, 1]$, if a constraint is put on Ω_m (or equivalently, s).

To accommodate further refinements, a correction term $w(x)$ can be added to the Hermite approximation. We require the value and first derivative of $w(x)$ to vanish at either end-point, so that it can be added to the Hermite approximation without altering the coefficients on the basis splines. One choice of $w(x)$ is made possible by a family of functions

$$w(x) = x^2(1-x)^2(ax+b+2a) \quad (8)$$

where a and b are adjustable parameters accounting for the deviation of the Hermite approximation from the true function. Other choices are possible, but we will begin with the simple case of equation (8).

By construction, the approximation described above has the property that the approximating function coincides with the true function at the end-points, x_* and 1 , up to the first derivative. But we note that the goal is to approximate equation (2) rather than equation (3). This suggests that the implicit requirement of the coincidence of function values at end-points could be unnecessarily strong. Alternatively, we may refrain from requiring the approximating function values to match the true ones. Instead, we only require the matching of first derivatives at $x = x_*$, and leave the end-point value at another free parameter. To summarize, we now have four free parameters that can be tuned: x_* , a and b , and the function value v_0 at $x = x_*$. The approximation to equation (6) can be expressed as

$$\tilde{\xi}(y) = v_0 H_0^{(0)}(y) + l \left[d_0 H_0^{(1)}(y) - 4H_1^{(1)}(y) \right] + w(y), \quad (9)$$

where

$$l = 1 - x_*, \quad y = \frac{x - x_*}{l}, \quad d_0 = \xi'(x_*),$$

and $H_i^{(j)}$ are the Hermite basis splines,

$$\begin{aligned} H_0^{(0)}(y) &= 2y^3 - 3y^2 + 1, \\ H_1^{(0)}(y) &= -2y^3 + 3y^2, \\ H_0^{(1)}(y) &= y^3 - 2y^2 + y, \\ H_1^{(1)}(y) &= y^3 - y^2. \end{aligned}$$

Following the approach in Pen99, we choose the objective function as the maximum relative error in d_L using the approximation ([Eq. 9]), with the restriction $0.2 \leq \Omega_m \leq 1$. Minimizing the objective function over the parameters, we obtain the best-fit $x_* = 0.40176$, $a = 1.62053$, $b = -6.34985$,

¹ See (Olver et al. 2010, Chapter 19), available online at <http://dlmf.nist.gov/19.29>

² We note in passing that the constant X in Pen99, equation (5) is identical to $1/A$. A typo was made therein, which should have been $X \equiv [\int_0^\infty du/\sqrt{u^4+u}]^{-1}$.

and $v_0 = 4.64111$. Substituting the numerical values into equation (9), we therefore construct an approximation polynomial

$$\tilde{\xi}(y) = 1.62053y^5 - 6.34985y^4 + 8.41443y^3 - 2.01328y^2 - 6.31293y + 4.64111. \quad (10)$$

Equation (10) is the main result of this section. With the parameters determined, the approximation to d_L can be computed using this formula with equations (2) and (6).

4 PERFORMANCE OF THE METHODS

In this section we proceed to assess the performance of the methods in Sections 2 and 3. The assessment is mainly done in terms of the accuracy and efficiency.

4.1 Accuracy

The first method can be used to yield highly accurate numerical approximation of d_L for vast ranges of z and the parameter s if we adopt the algorithm for R_F by Carlson (1979, 1994). Unlike the methods based on the evaluation of a closed approximation formula, the desired cutoff error can be prescribed to determine when the iterative computation of R_F terminates. In practice, we found that the prescription of relative error $\sim 10^{-16}$ can be achieved without suffering significant loss in the computation speed.

For the second method, we plot the distribution of the relative error of d_L in Figure 1. As can be seen from the figure, the second method remains an approximation at best. Under our choice of fitting parameters and range of Ω_m , the relative error in d_L is $\sim 0.5\%$. The major source of this error is contributed by $z < 0.1$. For $0.1 < z < 10$ our method is comparable with that of Pen99, and ours slightly outperforms it when z is larger.

4.2 Efficiency

Theoretically, the best-, worst-, and average-case temporal efficiencies for each method can be calculated or estimated by tracking every operation taken during the course of the computing. However, such a thorough analysis is beyond the scope of this paper. Instead, we empirically compare the running time of the computer programs using the two methods with those of Pen99 and WU10 under a controlled environment.

In Figure 2 we display the benchmark results of our methods compared with that of Pen99 and WU10. To simulate a “real-world” application of these methods, we create a sample of SNIa redshifts using the Supernova/Acceleration Probe (SNAP Collaboration 2004) fiducial redshift distribution containing 1998 redshift points distributed within $0.1 \leq z \leq 1.7$ (see Shafieloo et al. 2006, Table 1). Our sample satisfies the same distribution to the SNAP fiducial, but is 16 times as dense, i.e. with 31968 points in total. We have made custom implementations of the methods from Pen99, WU10, and our Method II in the C programming language, and uses the GNU Scientific Library (GSL³)

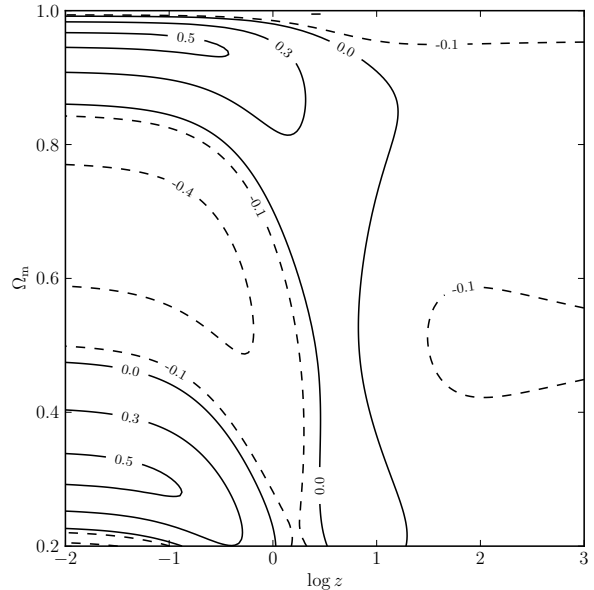


Figure 1. Contour plot for the distribution of the relative error in d_L using the approximation method in Section 3. Positive and negative values of the error are plotted in solid and dashed lines respectively. The “peaks” and “pits” in the left region of this figure ($z < 0.1$) dominate the global error.

implementation of the R_F algorithm in (Carlson & Notis 1981) for Method I. In our benchmark routine, the computing of d_L values from our redshift sample is performed for $\Omega_m = 0.3, 0.5, 0.7$, and 0.9 respectively, with each pass through the z sample repeated for 25 times (that sums up to a total of 3.2×10^6 evaluations of d_L). The benchmark itself is repeated for 2400 times.

To interpret Figure 2, we make two remarks. First, the execution time results were collected from the output of the **gprof** profiler⁴ and does not reflect the absolute time spent. It is only meaningful as a relative measure useful for comparing the speed of the codes with each other. Second, the results are dependent on our particular implementations as well as the computing environment. This is evident if our Figure 2 is compared with Figure 4 in WU10 that shows a reversed result for the speeds of the two methods in Pen99 and WU10.

5 DISCUSSION

As Figure 2 suggests, both methods proposed in this paper is slower than the Pen99 method. However, Method I is a very reasonable trade-off between an enormous gain in accuracy and small loss of efficiency. With Method I one does not need to resort to the numerical quadrature for the same level of accuracy.

Method I can be extended to cover the Λ CDM model

³ <http://www.gnu.org/software/gsl/>

⁴ <http://www.gnu.org/software/binutils/>

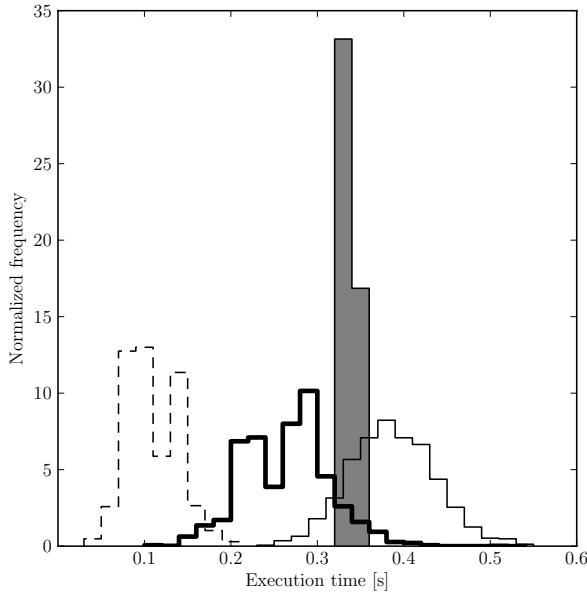


Figure 2. Histograms of the benchmark results for the four methods. Dotted, thick solid, shaded, and thin solid histograms represent the execution timings of codes implementing Pen99, our Method I, Method II, and WU10 respectively. Each histogram is normalized so that the total probability (the area enclosed under the boundaries) is unity.

with a curvature term Ω_k , because in that case the equivalent of equation (1) assumes the form

$$\frac{dL}{c/H_0} = \frac{1+z}{\sqrt{|\Omega_k|}} \text{sinn} \left[\sqrt{|\Omega_k|} \int_0^z \frac{dt}{E(t; \Omega_m, \Omega_k)} \right] \quad (11)$$

where

$$E(z; \Omega_m, \Omega_k) = \sqrt{\Omega_m(1+z)^3 + \Omega_k(1+z)^2 + \Omega_\Lambda}$$

is the expansion rate ($\Omega_\Lambda = 1 - \Omega_m - \Omega_k$), and

$$\text{sinn}(x) = \begin{cases} \sin(x), & \Omega_k < 0; \\ x, & \Omega_k = 0; \\ \sinh(x), & \Omega_k > 0. \end{cases}$$

The integral in equation (11) is also an elliptic integral and can be reduced to R_F accordingly. This is potentially useful for the analysis of future SNIa data, because it has been suggested that the spacetime curvature should not be ignored in the probe of dark energy using luminosity distance data (Clarkson et al. 2007; Öztaş et al. 2008).

In contrast, Method II may not be as promising, because in its current form the accuracy does not outperform that of Pen99. However, the idea behind the method may be useful when extending to alternative cosmological models (for example, dynamical dark energy) which may not be reduced to the applicable scenarios of Method I. In the description of this method we have left some arbitrariness unjustified, notably the particular choice of the singularity-removing transformation (Eq. [6]), the parameterization of the correction term (Eq. [8]), and the very choice of Hermite basis splines. Alternative choices of them may be adopted to

generate better approximations, for instance, the use of low-order Hermite-Birkhoff interpolation⁵ to selectively choose the point $x \in [0, 1]$ near which the derivative information of the true function is to be best preserved. Moreover, our Method II uses only elementary numerical operations, while in WU10 the numerical logarithm is extensively used.

ACKNOWLEDGEMENTS

De-Zi Liu would like to thank Fang-Fang Zhu and Mao-Sheng Xiang for their kind help. This work was supported by the National Science Foundation of China (Grants No. 10473002), the Ministry of Science and Technology National Basic Science program (project 973) under grant No. 2009CB24901, the Fundamental Research Funds for the Central Universities.

References

- Abramowitz M., Stegun I. A., eds, 1972, Handbook of Mathematical Functions. U.S. Gov. Printing Office, Washington, D.C.
- Carlson B. C., 1979, Numerische Mathematik, 33, 1
- Carlson B. C., 1994, preprint, arXiv:math/9409227
- Carlson B. C., Notis E. M., 1981, ACM Transactions on Mathematical Software, 7, 398
- Clarkson C., Cortès M., Bassett B., 2007, J. Cosmol. Astropart. Phys., 8, 11
- Kress R., 1998, Numerical Analysis. Springer-Verlag, New York
- Olver F. W. J., Lozier D. W., Boisvert R. F., Clark W. C., eds, 2010, NIST Handbook of Mathematical Functions. Cambridge Univ. Press, New York
- Öztaş A. M., Smith M. L., Paul J., 2008, International Journal of Theoretical Physics, 47, 2464
- Pen U.-L., 1999, ApJS, 120, 49
- Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., 2007, Numerical Recipes: The Art of Scientific Computing, 3 edn. Cambridge Univ. Press, Cambridge, UK
- Shafieloo A., Alam U., Sahni V., Starobinsky A. A., 2006, MNRAS, 366, 1081
- Sharma A., Prasad J., 1968, SIAM Journal on Numerical Analysis, 5, 864
- SNAP Collaboration 2004, preprint, arXiv:astro-ph/0405232
- Whittaker E. T., Watson G. N., 1969, A Course of Modern Analysis, 4 edn. Cambridge Univ. Press, New York
- Wickramasinghe T., Ukwatta T. N., 2010, MNRAS, 406, 548

⁵ See (Sharma & Prasad 1968) for the general theory, and Problem 8.9 of (Kress 1998) for a low-order example.